We claim:

1.   A method for creating a binary tree from a list of elements, wherein the list includes left side groupings and right side groupings, the method comprising: inserting left side descendent nodes into the binary tree by successively finding the median of each left side grouping and linking each found median to the previous median; inserting right side descendent nodes into the binary tree by successively finding the median of each right side grouping and linking each found median to the previous median; and processing left side groupings in preference to right side groupings.

2.   A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 1.

3.   The method of claim 1, wherein each element in the list includes a pointer to a corresponding node of a plurality of nodes in a partially assembled binary tree, wherein each node has a left child pointer, and wherein inserting the left side nodes further comprises assigning a value to the left side pointer of at least one of the nodes.

4.   The method of claim 1, wherein each element in the list includes a pointer to a corresponding node of a plurality of nodes in a partially assembled binary tree, wherein each node has a right child pointer, and wherein inserting the right side nodes further comprises assigning a value to the right side pointer of at least one of the nodes.

5.   The method of claim 1, wherein inserting the left side descendent nodes comprises inserted the left side descendent nodes into a partially assembled version of the binary tree, wherein inserting the right side descendent nodes comprises inserted the right side descendent nodes into the partially assembled version of the binary tree, and wherein the list is a linked list that acts as a wrapper around the partially assembled version of the binary tree.

6.   The method of claim 1, wherein the elements of the list include logged events.

7.   The method of claim 1, wherein the elements of the list include data representing the number of times one or more threads of execution have passed through one or more code modules.

8.   The method of claim 1, wherein the inserted right and left descendant nodes include data representing the number of times one or more threads of execution have passed through one or more code modules.

9.   The method of claim 1, wherein the inserted right and left descendant nodes include one or more pointers to data representing the number of times one or more threads of execution have passed through one or more code modules.

10.   The method of claim 1, wherein the list is an ordered linked list.

13

11.   A method for creating a binary tree from a list of elements, wherein the list includes left side groupings and right side groupings, the method comprising:

(a) successively subdividing the left side groupings of the list and linking each successive median element with the previous median element, thereby creating left side descendent nodes in the binary tree;

(b) once each left side grouping has been exhausted as a result of step (a), stepping back up the tree through each successive ancestor node until reaching an element having right side groupings in the list, and, upon reaching an element having a right side grouping in the list, proceeding to step (c);

(c) subdividing the right side groupings and linking the median element of the right side grouping with the element reached in step (b), thereby creating a right side descendent of the tree;

(d) if the right side descendent of step (c) has a left side grouping in the list, repeating step (a) for the left side grouping; and

(e) if the right side descendent of step (c) has no left side groupings, but has a right side grouping, repeating step (c) for the right side grouping.

12.   A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 11.

13.   A method for creating a binary tree from a list of elements, wherein the list includes left side groupings and right side groupings, the method comprising:

(a) determining if there are elements to the left of a parent element;

14

(b) if there are no elements to the left of the parent element, proceeding to step (f);

(c) for the elements that are to the left of the parent element, finding a median element;

(d) linking the median element of step (c) to the parent element so that the median

5   element is the child of the parent element;

(e) repeating steps (b) and (c), wherein the child element of step (d) is now treated as the parent element in steps (b) and (c);

(f) locating the next element up on the tree that has elements to the right of it and treating the element as a parent element in step (g);

10   (g) finding the median element of the elements to the right of the parent element from step (f);

(h) linking the median element of step (g) to the parent element of step (g), wherein the median element is the child of the parent; and

(i) repeating steps (b) through (e), wherein the child element of step (h) is treated as
15   the parent element in step (b).

14.   A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 13.

20   15.   A method for creating a binary tree from a list of elements, wherein the list includes left side groupings and right side groupings, the method comprising: inserting right side descendent nodes into the binary tree by successively finding the median of each right side grouping and linking each found median to the previous median; inserting left side

15

descendent nodes into the binary tree by successively finding the median of each left side grouping and linking each found median to the previous median; and processing right side groupings in preference to left side groupings.

5   16.   A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 15.

17.   The method of claim 15, wherein each element in the list includes a pointer to a corresponding node of a plurality of nodes in a partially assembled binary tree, wherein each node has a right child pointer, and wherein inserting the right side nodes further comprises assigning a value to the right side pointer of at least one of the nodes.

18.   The method of claim 15, wherein each element in the list includes a pointer to a corresponding node of a plurality of nodes in a partially assembled binary tree, wherein each node has a left child pointer, and wherein inserting the left side nodes further comprises assigning a value to the left side pointer of at least one of the nodes.

19.   The method of claim 15, wherein inserting the right side descendent nodes comprises inserted the right side descendent nodes into a partially assembled version of the
20   binary tree, wherein inserting the left side descendent nodes comprises inserted the left side descendent nodes into the partially assembled version of the binary tree, and wherein the list is a linked list that acts as a wrapper around the partially assembled version of the binary tree.

20.   The method of claim 15, wherein the elements of the list include logged events.

5   21.   The method of claim 15, wherein the elements of the list include data representing the number of times one or more threads of execution have passed through one or more code modules.

22.   The method of claim 15, wherein the inserted right and left descendant nodes

10   include data representing the number of times one or more threads of execution have passed through one or more code modules.

23.   The method of claim 15, wherein the inserted right and left descendant nodes include one or more pointers to data representing the number of times one or more threads of

15   execution have passed through one or more code modules.

24.   The method of claim 15, wherein the list is an ordered linked list.

25.   A method for creating a binary tree from a list of elements, wherein the list

20   includes right side groupings and left side groupings, the method comprising:

(a) successively subdividing the right side groupings of the list and linking each successive median element with the previous median element, thereby creating right side descendent nodes in the binary tree;

17

(b) once each right side grouping has been exhausted as a result of step (a), stepping

back up the tree through each successive ancestor node until reaching an element having left

side groupings in the list, and, upon reaching an element having a left side grouping in the

list, proceeding to step (c);

5   (c) subdividing the left side groupings and linking the median element of the left side

grouping with the element reached in step (b), thereby creating a left side descendent of the

tree;

(d) if the left side descendent of step (c) has a right side grouping in the list, repeating

step (a) for the right side grouping; and

10   (e) if the left side descendent of step (c) has no right side groupings, but has a left side

grouping, repeating step (c) for the left side grouping.

26.   A computer-readable medium having stored thereon computer-executable

instructions for performing the method of claim 25.

15   27.   A method for creating a binary tree from a list of elements, wherein the list

includes right side groupings and left side groupings, the method comprising:

(a) determining if there are elements to the right of a parent element;

(b) if there are no elements to the right of the parent element, proceeding to step (f);

20   (c) for the elements that are to the right of the parent element, finding a median

element;

(d) linking the median element of step (c) to the parent element so that the median

element is the child of the parent element;

(e) repeating steps (b) and (c), wherein the child element of step (d) is now treated as the parent element in steps (b) and (c);

(f) locating the next element up on the tree that has elements to the left of it and treating the element as a parent element in step (g);

5   (g) finding the median element of the elements to the left of the parent element from step (f);

(h) linking the median element of step (g) to the parent element of step (g), wherein the median element is the child of the parent; and

(i) repeating steps (b) through (e), wherein the child element of step (h) is treated as

10   the parent element in step (b).

28.   A computer-readable medium having stored thereon computer-executable instructions for performing the method of claim 27.